

University of Groningen

Matrix renumbering ILU

Botta, E. F. F.; Wubs, F.W.

Published in:
SIAM Journal on Matrix Analysis and Applications

DOI:
[10.1137/S0895479897319301](https://doi.org/10.1137/S0895479897319301)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
1999

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Botta, E. F. F., & Wubs, F. W. (1999). Matrix renumbering ILU: An effective algebraic multilevel ILU preconditioner for sparse matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(4), 1007-1026. <https://doi.org/10.1137/S0895479897319301>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

MATRIX RENUMBERING ILU: AN EFFECTIVE ALGEBRAIC MULTILEVEL ILU PRECONDITIONER FOR SPARSE MATRICES*

E. F. F. BOTTA[†] AND F. W. WUBS[†]

Abstract. In this paper a multilevel-like ILU preconditioner is introduced. The ILU factorization generates its own ordering during the elimination process. Both ordering and dropping depend on the size of the entries. The method can handle structured and unstructured problems. Results are presented for some important classes of matrices and for several well-known test examples. The results illustrate the efficiency of the method and show in several cases near grid independent convergence.

Key words. multilevel methods, preconditioning, ILU, dropping strategies, Krylov-subspace methods

AMS subject classifications. 65F10, 65N06

PII. S0895479897319301

1. Introduction. Solving large sparse systems of equations continues to be a major research area. This attention is caused by the fact that solving such equations forms the bottleneck in many practical problems. For really large systems direct methods become too expensive in CPU time and storage requirements, and therefore an iterative approach is needed. In particular the use of preconditioned CG-type methods has proved to be very competitive. It is also widely recognized that the quality of the preconditioner determines the success of the iterative method. With a proper preconditioner the choice of the CG-like accelerator is not that critical.

The preconditioner presented in this paper is a special multilevel-like incomplete factorization. In this introduction we briefly describe the various incomplete decomposition approaches available today and their relation to the approach presented here.

The history of ILU factorizations is amongst others described in [15]. Moreover, historical notes are to be found in the textbooks of Axelsson [1], Hackbusch [31], and Saad [50]. The first roots of the approach lie in the 1960s [13, 42, 43] and since then the method has become applicable to a wide class of problems. Furthermore, analyses for important classes of matrices could be made. Today, ILU factorizations are an important tool for solving large-scale problems.

Classical ILU approach. The classical approach is to allow only fill entries in the L and U factors, where the original matrix A has nonzeros. This simple approach allows for a very efficient implementation by Eisenstat [25] and is still very popular.

As observed by Dupont, Kendall, and Rachford [23], an important improvement in the convergence of the classical approach can be obtained by lumping the dropped elements onto the diagonal. With this modification, the factorization, called MILU, is made exact for a constant vector. For a more general matrix A Gustafsson [30] found a similar result. For many second-order elliptic problems, the preconditioning with the classical ILU gives asymptotically the same condition number as with diagonal scaling, i.e., $O(h^{-2})$. After this simple modification this improves to $O(h^{-1})$. For M-matrices the existence of ILU factorizations can be proved [35], but this is not the case

*Received by the editors March 31, 1997; accepted for publication (in revised form) by R. Freund January 19, 1998; published electronically July 9, 1999.

<http://www.siam.org/journals/simax/20-4/31930.html>

[†]Department of Mathematics, University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands (E.F.F.Botta@math.rug.nl, F.W.Wubs@math.rug.nl).

for MILU factorizations. Here, the factorization may break down [24]. The subject of existence is studied by Beauwens and Quenon [5] and Notay [38, 39]. Relaxed forms of MILU have been introduced to prevent the problem of breakdown and bad conditioning [40]. Problems occur at positions in the matrix where due to lumping the diagonal becomes very small.

The classical approach is an example of *drop-by-position*, which can be generalized by also allowing fill at other user-specified positions in the L and U factors. In general it is difficult to determine where to allow fill. Hence, matrix-dependent approaches are to be favored.

Matrix-dependent approaches. One may distinguish two approaches: one in which only the nonzero structure of the original matrix determines the fill pattern, and another in which the values of the coefficients also are involved.

An example of the first approach is the one based on the level of fill. The levels are defined recursively. Entries belonging to the original nonzero structure of the matrix are defined to have a level of fill of zero. Fills in the LU factors caused by entries of level k have level $k + 1$. For many problems the size of elements decreases with the level number and in practice the number of levels is kept low (see [50, section 10.3.3]). This approach appears to be rather successful [17]. However, there can still be a lot of fill of which most entries are very small.

In the second approach, called *drop-by-size* method (or incomplete factorization by value, e.g., [1]) such small elements are dropped according to a dropping rule. Let us briefly review some of the drop-by-size strategies.

In the ILU method, proposed by Saad [50, section 10.4.1], the factors L and U are constructed row by row, with L unit lower triangular. When constructing row i , first the 2-norm of the i th row of the original matrix is calculated. Multiplied by a user specified tolerance, this is used as a drop tolerance for this row and the multipliers used in the construction of this row. The fill in L and U is further limited by keeping, besides the diagonal, only the p largest elements in the L and U parts of the row.

Axelsson and Munksgaard [3] and Axelsson [1] follow the standard construction of the L and U factors. Elements in the part of the matrix still to be factorized at step k of the construction are dropped if

$$|a_{ij}^{(k)}| \leq \varepsilon |a_{ii}^{(k)} a_{jj}^{(k)}|^{1/2}.$$

For block matrices the dropping condition is generalized to

$$\|A_{ij}^{(k)}\| \leq \varepsilon \{\|A_{ii}^{(k)}\| \|A_{jj}^{(k)}\|\}^{1/2}.$$

This approach is strongly based on symmetric positive definite systems.

D'Azevedo, Forsyth, and Tang [18] base the dropping rule on the maximum values in rows i and j of the original matrix, i.e.,

$$|a_{ij}^{(k)}| < \varepsilon \min(\|a_{i*}\|_{\infty}, \|a_{j*}\|_{\infty}).$$

This condition is applied successfully to nonsymmetric problems. Nevertheless, one should be careful when the maximum does not occur at the diagonal position and as a result large multipliers may come across.

Ordering strategies. In any (I)LU factorization the ordering may have a significant impact on the amount of fill. Finding an optimal ordering is difficult (NP-hard), but over the years several successful heuristics have evolved. Some well-known examples are (reverse) Cuthill–McKee, minimum degree, and nested dissection (see,

e.g., [21]). For a variety of orderings the effect on preconditioned CGs has been studied by Duff and Meurant [22]. It turns out that orderings can have a dramatic influence on the convergence and that the norm of the residual matrix $R = A - LU$ is useful as an estimate for the quality of the preconditioning. It appears that for ILU(0) the simple row ordering and the spiral ordering perform very well.

When more drop-tolerance controlled fill is allowed, the situation is different. Now the red-black ordering and the alternating diagonal ordering are very competitive. In the matrix renumbering ILU (MRILU) method, described in section 2, this type of ordering is generated automatically for each new Schur complement during the elimination process.

In D'Azevedo, Forsyth, and Tang [18] a near-optimal ordering is constructed during the elimination process. The basic idea is to choose in each step of a level-of-fill approach the next pivot node that minimizes the Frobenius norm of a discarded fill matrix. Hence in some sense the norm of the residual matrix is minimized. For ILU(0) on a standard Laplace problem this leads to an ordering very similar to the spiral ordering and with ILU(1) a generalized red-black ordering evolves. This agrees nicely with the results of Duff and Meurant [22], who also show that in case of anisotropy the ordering becomes even more critical. In such cases the method designed by D'Azevedo, Forsyth, and Tang constructs an ordering that is different from the isotropic case, leading to a better convergence. An interesting analysis for modified block incomplete factorizations is given by Magolu [36]

Repeated red-black orderings have been studied extensively. Brand [12], and separately Axelsson and Eijkhout [2], analyzed this approach. Notay and Ould Amar [41] showed recently a bound for the condition number of the preconditioned matrix $O(N^{0.153})$, where N is the total number of unknowns. Hence the condition number comes close to the optimal value one. In [8] we considered a fixed red-black ordering and combined this with a dropping strategy. It appeared that grid-independent convergence is possible if used in a CG process.

MRILU as presented in this paper determines the ordering by itself. At every block step in the process we determine an independent set, i.e., a set of unknowns not directly connected in the graph. The unknowns associated with an independent set can be eliminated simultaneously. This concept is stretched to the case of a weak coupling. This is very similar to the work done by Saad [50] and Saad and Zhang [51] in ILU with multielimination (ILUM); however, there each elimination is exact and at a later stage the resulting system is solved by a simple iteration scheme. It appears that in our approach again grid-independent convergence is possible even for unstructured problems. An interesting side effect of red-black orderings is their attractiveness for implementation on supercomputers. This was the basic argument for Saad to develop ILUM. In [10] we constructed such an implementation for a shared-memory parallel computer using block slicing to distribute the work over the processors and jagged-diagonal storage to speed up the vector processing.

The above orderings ask for a matrix close to an M-matrix and are not very suitable for many other matrices. The (Navier-) Stokes problem, for example, leads to matrices that differ largely from an M-matrix. The Stokes problem can be written as a system with a symmetric indefinite matrix. It has negative and positive eigenvalues. The Navier-Stokes equation is a perturbation of the Stokes equation but is not symmetric. Another example is the convection-diffusion equation with strong convection and discretized by central differences. In [27, sect. 4.4] an overview of direct methods for this type of problem is given.

The symmetric structure of the matrix is destroyed by partial or complete pivoting and therefore we always employ diagonal pivoting (reordering). A form relevant to our approach is the diagonal pivoting method of Bunch and Parlett [14]. In this method the notion of a pivot is extended to 2×2 blocks that are used when, due to large elements outside the diagonal, choosing a 1×1 pivot on the diagonal is no longer stable.

Relation to algebraic multigrid. Grid-independent convergence has become almost synonymous with multigrid. The algorithms in the multilevel world can be classified into two groups: parallel subspace correction (PSC) and successive subspace correction (SSC) [61]. Some papers use the corresponding terminology from domain decomposition approaches: additive Schwarz and multiplicative Schwarz methods. The two types are similar to the Jacobi and Gauss–Seidel methods. The first is more suited for parallel computations whereas the second is more attractive on sequential computers. The classical V-cycle multigrid is an SSC method (see [61]), whereas the Bramble–Pasciak–Xu (BPX) method [11] is a PSC method. In the classical approaches the sequence of nested subspaces is given and together with the corresponding orthogonal projections the coarse grid matrices are constructed by means of the Galerkin approach. One way to improve the robustness of this classical approach is to use matrix-dependent projections [19, 44, 45]. As a next step the subspaces can be determined by the matrix. This resulted in the so-called algebraic multigrid (AMG). Although these steps were first done in the multiplicative context they can also be applied in the additive form (see [28]).

The approach we follow is connected with AMG as introduced by Ruge and Stüben [47]. In their approach two steps in particular are important. The first step is the selection of coarse grid points. The coarse grid points should be distributed uniformly over the grid such that the matrix can be ordered in a 2×2 block partition with the property that the coupling between points in A_{22} is weak. The second step is the construction of an interpolation formula, again using the matrix entries. Once this interpolation operator with a weighting of unity is settled, the coarse grid operator can be constructed by a Galerkin approach. Thereafter the construction is started again at the coarse grid. In the iteration process smoothing at each level has to be applied to get rid of high-frequency errors. In [16] an AMG method is presented that can handle nonsymmetric problems. Here, a special interpolation formula is derived that can handle positive off-diagonal elements and is accurate for linear functions. Moreover, the coarse grid operator is computed more accurately by an approximate elimination. This results in a rather robust method.

In a certain sense our approach may be viewed as such an AMG method. However, we don't use smoothing and the prolongation and restriction operators appear in a natural way during the decomposition. As with AMG only one algorithm can handle various problems, which is not the case for the geometrically oriented multigrid algorithms.

In the following we will assume that the matrices originate from a partial differential equation (PDE) without any specific scaling of columns. If possible we start with a symmetric matrix; otherwise a row scaling is performed such that a reasonable degree of normality is obtained [17]. In the case of a *system* of PDEs the unknowns should be properly scaled by the user. Note that dropping strategies are sensitive to scaling. Scaling is a difficult job and can best proceed on a problem-by-problem basis; see [27] for further discussion.

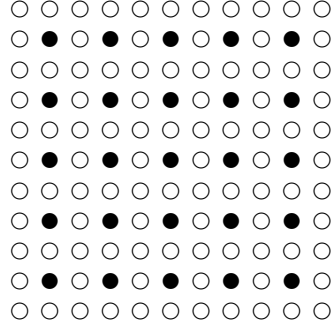


FIG. 1. Hierarchical multigrid ordering in NGILU.

2. MRILU. MRILU generalizes an earlier method called nested grids ILU (NGILU) [8]. For a better understanding of MRILU we will therefore start with a short description of this method. NGILU uses a multilevel approach as in multigrid and combines a hierarchical multigrid ordering of the unknowns with an easy-to-construct ILU factorization using a special drop tolerance technique. The construction of the incomplete factorization is illustrated by a brief discussion of the first reduction step. Suppose we start with the system $Sx = b$ on the grid given in Figure 1. On this grid we distinguish between points/unknowns \circ and \bullet , where the dots are the points from a next-coarser grid. This allows the system to be partitioned as

$$(1) \quad \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} \circ \\ \bullet \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

For many problems it can be shown that S_{11} is well-conditioned. Think of S_{11} as the matrix that remains when the unknowns in the dots are prescribed. In that case we will get rid of the smallest eigenvalues and the corresponding smooth eigenvectors (low frequencies). For this well-conditioned S_{11} we can easily construct a sparse incomplete factorization \tilde{S}_{11} and use it with a drop tolerance ε to construct the incomplete factorization

$$\tilde{S} = \begin{pmatrix} \tilde{S}_{11} & \tilde{S}_{12} \\ \tilde{S}_{21} & \tilde{S}_{22} \end{pmatrix} = \begin{pmatrix} \tilde{S}_{11} & O \\ \tilde{S}_{21} & S^{(2)} \end{pmatrix} \begin{pmatrix} I & \tilde{S}_{11}^{-1} \tilde{S}_{12} \\ O & I \end{pmatrix}.$$

Here the Schur complement

$$S^{(2)} = \tilde{S}_{22} - \tilde{S}_{21} \tilde{S}_{11}^{-1} \tilde{S}_{12}$$

corresponds with the reduced system obtained after elimination of the unknowns in the circles. Now we repeat the process for this reduced system on the coarser grid given by the dots in Figure 1. We continue this approach until we obtain a Schur complement small enough to be solved with some standard method and we finally arrive at the incomplete factorization

$$A = LU + R,$$

where R is called the residual matrix. To obtain grid-independent convergence, it is essential that the drop tolerance ε decreases as we go to higher levels (normally by a factor of 4 or 5 in two dimensions). If we consider, just as an illustration, the

extreme case that the residual matrix has only nonzeros in the first left upper part R_{11} , the vector Rx contains only components of the first level. With this type of preconditioner all low-frequency errors are eliminated immediately and the iterative method has only to remove high-frequency errors with a wavelength in the order of the mesh size.

This approach can be combined with Gustafsson's modification. It is also advantageous to use an appropriate ordering of the unknowns within each level. For a five-point stencil we can use a red-black ordering. This offers the possibility of starting in the first step with an exact reduction of about half of the unknowns. This will always be the starting point and the preconditioning and acceleration are actually performed for this reduced problem.

Some results of NGILU are given in the next section. For many problems NGILU leads to fast, smooth, and (almost) grid-independent convergence. However, shortcomings of NGILU are the need for a structured grid and, for supercomputers, the sequential nature of the forward and backward substitution when solving $LUx = b$. It also turns out that the method does not perform well on stretched grids, a property it shares with a number of popular iterative methods [9].

The problem with NGILU is that the numbering is based entirely on the grid and does not take into account the size of the nonzero elements of the matrix. For an efficient factorization it is crucial that the upper left blocks S_{11} are always well-conditioned. The difficulties with the stretched grid are caused by the fact that there this condition is violated.

These shortcomings have been overcome in MRILU, the generalized version of NGILU. Here the renumbering is not made beforehand on the basis of the underlying grid or the sparsity pattern [21] but is determined during the construction of an incomplete block factorization using the sparsity pattern and the magnitude of the elements. The method is related to ILUM of Saad [48, 49]. During the factorization we guarantee by construction that the diagonal blocks to be inverted are strongly diagonally dominant, i.e., the coefficients of S_{11} satisfy

$$\sum_{i \neq k} |s_{ik}| \leq \varepsilon |s_{ii}| \text{ with } \varepsilon < 1.$$

By taking ε small enough, we can approximate S_{11} by a diagonal matrix. This not only simplifies the construction of the next Schur complement but also leads to more potential parallelism. The unknowns belonging to S_{11} can be eliminated simultaneously. This also means that the ordering within this level is no longer relevant.

For general matrices the renumbering can be constructed by a greedy algorithm. In the implementation we keep track of the absolute sum of columns belonging to the points selected for S_{11} . For each following candidate we now can easily verify whether it can be added to the (near) independent set selected thus far.

In this way each Schur complement S is partitioned as

$$\begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

and the strongly diagonally dominant S_{11} is replaced by a diagonal matrix.

To limit the number of nonzeros, small elements will be dropped during the construction. It is difficult to study the existence and stability of ILU factorizations for general matrices and even more difficult to study the effect of dropping on the

eigenvalues of the preconditioned system. Therefore dropping criteria as used in practice are based on heuristics.

Dropping an element s_{ij} and, in the modified approach, compensating for this dropping by adding s_{ij} to s_{ii} means a modification of row i and column j . Whether this dropping is acceptable is usually measured in terms of s_{ii} and s_{jj} . For stretched grids or strongly varying coefficients the diagonal of the next Schur complement can be significantly smaller than the original diagonal of S_{22} . Relatively small modifications of the diagonal in S_{22} may be large compared to this new diagonal. The diagonal of the next Schur complement can become even smaller in the steps to follow. This is the main reason that the factorization can easily break down (cf. section 1). To anticipate these problems we would prefer dropping criteria based on the nontrivial diagonal D in the final factorization. The diagonal of S_{11} becomes a (slightly modified) part of D and this explains that, just as with NGILU, dropping elements outside S_{11} is much more critical. We limit the dropping outside S_{11} to S_{12} and S_{21} and approximate D within S_{22} by the diagonal of the next Schur complement (temporarily calculated without dropping outside S_{11}). For dropping we now demand that on each row or column i the sum of the absolute values of all discarded elements, including the values belonging to earlier levels, is smaller than $\varepsilon|d_i|$, with d_i the “updated” diagonal element on row i . When dropping entries in S_{21} on a certain row, that same row might also be accessed for dropping on higher levels. Therefore we limit the row space for dropping within S_{21} even further and multiply the remaining space by the number of columns in S_{21} divided by the dimension of S . Finally, the available space for lumping on rows of S_{21} should not be consumed too fast and therefore only entries smaller than a quarter of this space are dropped. Similar restrictions are made for the dropping within columns of S_{12} .

For many problems it turns out that the d_i become more or less constant during the block factorization; see also [33]. This offers the possibility to simplify the dropping strategy and use a constant d_i just from the start. In the next section we refer to it as the simplified dropping strategy.

The above approach can be generalized to a block approach. To decide whether entries in a certain block row can be dropped, we multiply this block row from the left with the inverse of the corresponding diagonal block. This can be simplified somewhat by considering only the absolute maximum in each column of this inverse. The dropping in columns can be handled in a similar way using multiplication from the right.

Theoretical results for this type of preconditioner or the related AMG-based approach are rare, as is also observed in [28]. In general it is restricted to existence of the decomposition and analysis of special cases. For the case of a symmetric diagonally dominant M -matrix some theoretical results with respect to the NGILU method can be found in [56, 8]. For MRILU it is shown in [7] that on all levels the constructed Schur complements are again symmetric diagonally dominant M -matrices. Similar results can be found in [4, 46].

3. Results. In the first two problems the typical behavior of NGILU is illustrated by a comparison to other standard methods. In the variety of problems that follows, the results of MRILU are compared with those of several other methods. We will focus our attention on the solution process.

Of course the cost for solving depends upon the quality of the preconditioner, hence on the cost for its construction. For one of the examples described in section 3.4, i.e., the Poisson equation on a finite element mesh with 37,791 unknowns, detailed

TABLE 1
Results on a finite element mesh for various values of ε .

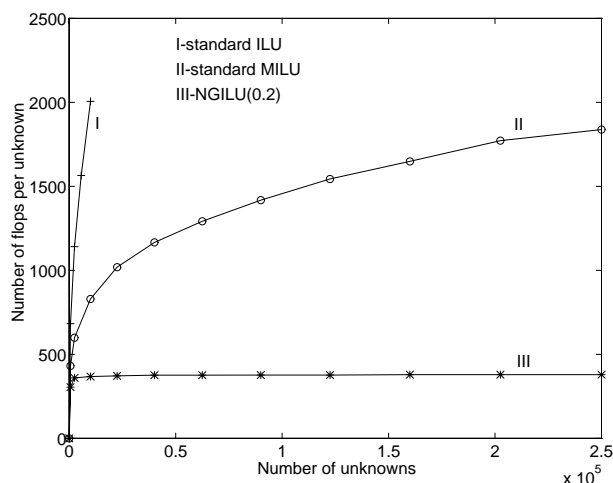
ε	Flops prec.	Flops solving	Flops	Rel. fill	Levels	It.
0.7	152	1699	1851	0.77	12	45
0.6	174	1057	1231	0.86	13	27
0.5	194	727	921	0.95	14	18
0.4	223	546	769	1.1	16	13
0.3	284	501	785	1.3	17	11
0.2	368	441	809	1.6	19	9
0.15	481	424	905	1.9	21	8
0.1	764	363	1127	2.4	24	6
0.03	1684	311	1995	3.6	35	4
0.01	3279	293	3572	5.1	46	3
0.003	6231	246	6477	6.9	59	2
0.001	10221	296	10517	8.7	69	2
0	55562	749	56311	54.0		1

information about what happens when going from a rough preconditioner ($\varepsilon = 0.8$) toward a complete factorization ($\varepsilon = 0$) is presented in Table 1.

The table shows (in flops per unknown) the cost for the construction of the preconditioner, the cost for solving, and the total cost. It further shows the fill (number of nonzeros in the incomplete LU factorization divided by the nonzeros in the original system), the number of levels, and, finally, the number of CG iterations needed for a decrease in the 2-norm of the residual of the preconditioned system by at least a factor of 10^6 . The results for ($\varepsilon = 0$) are from a standard Cholesky factorization. The behavior shown in Table 1 is typical for MRILU. The results presented in this section are obtained for a choice of ε such that the cost for construction of the preconditioner and the cost for solving are of the same order. Note that this choice is not very critical with respect to the total number of flops. The number of iterations is normally of the order of 10. The presented results for solving can be somewhat improved by using a smaller value of ε . This certainly can be done in time-dependent or nonlinear systems where the preconditioner can be used several times and the cost of its construction is therefore amortized over several solves. For more detailed information with respect to the costs and a comparison with other methods we refer to [6]. Unless denoted otherwise we will use Bi-CGSTAB as accelerator.

3.1. Poisson equation on a uniform grid. For a Poisson equation with constant coefficients discretized on a uniform grid it may be expected from the discussion in the previous section that the hierarchical multigrid ordering will work well. For Neumann boundary conditions on all sides of the unit square, the results for a number of standard methods and NGILU are given in Figure 2. The figure shows the number of flops per unknown necessary to improve the preconditioned residual by six digits versus the number of unknowns. In modified incomplete Cholesky CG (MICCG) small perturbations are applied to the diagonal as described in [32], i.e., before the factorization all diagonal elements are multiplied by $1 + \zeta h^2$. The choice $\zeta = 10$ is almost optimal in the present case. With standard incomplete Cholesky CG (ICCG) the number of flops per unknown grows very strongly with the problem size. Perturbed MICCG does a much better job, but NGILU is by far the best and shows grid independent convergence.

For this simple standard problem special-purpose solvers, e.g., solvers based on FFT or cyclic reduction, are an order of magnitude faster than NGILU(0.2); see [6]. Methods such as the BPX method [11] or the multilevel filtering technique [34], which

FIG. 2. Neumann problem $-\Delta u = f$ on a uniform grid.

are designed for constant coefficient problems (but also can operate on nonconstant coefficient cases), work well on this problem. From [34] we took the result of the standard BPX method for the corresponding problem with 12,000 unknowns (a point very near the left of our plot). To gain 6 digits 750 flops are needed, about twice as many as for NGILU(0.2). According to the bound for the condition number of the preconditioned matrix $O(\log h^{-1})$, this gap will increase with the number of unknowns. For this problem the multilevel filtering technique (an improvement of the BPX method) using an appropriate filter is about as efficient as NGILU(0.2).

3.2. An aquifer problem. As a less trivial example we take a simplified aquifer problem as described in [57]. The nonsymmetric system of linear equations is a result of the discretization of the following steady convection-diffusion equation:

$$-\frac{\partial}{\partial x} \left(a(x, y) \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left(a(x, y) \frac{\partial u}{\partial y} \right) + 2e^{2(x^2+y^2)} \frac{\partial u}{\partial x} = \begin{cases} 100 & \text{in center,} \\ 0 & \text{elsewhere.} \end{cases}$$

The diffusion equation coefficient function $a(x, y)$ and the Dirichlet boundary conditions are shown in Figure 3. Figure 4 shows the convergence behavior of Bi-CGSTAB on a 201×201 grid for ILU, MILU, and NGILU. The standard MILU factorization breaks down in this case. It is essential to use MILU with a small drop tolerance [55]. Note the smooth convergence behavior of NGILU, which is very favorable; see [57, 53, 52].

3.3. Poisson equation on a severely stretched grid. Investigating the flow in a driven cavity [58] leads to a solution where a lot of action occurs near the boundaries.¹ To get enough resolution near the boundaries, a grid is used, as given in Figure 5. Here we consider the solution of the homogeneous Poisson equation with zero Neumann boundary conditions on an exponentially stretched grid, where the ratio of maximum and minimum mesh size is given by $h_{max}/h_{min} = 100$. The results are obtained for a nonzero starting vector and the stopping criterion $u_{max}^{(n)} - u_{min}^{(n)} <$

¹See <http://www.math.rug.nl/~veldman/cfd-gallery.html>.

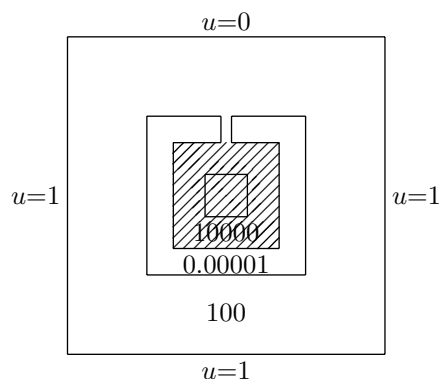
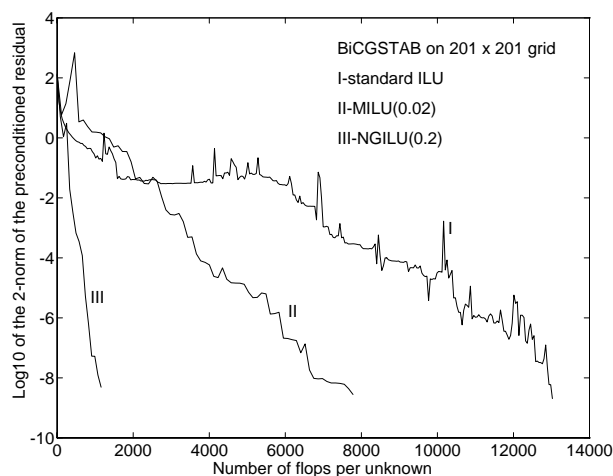
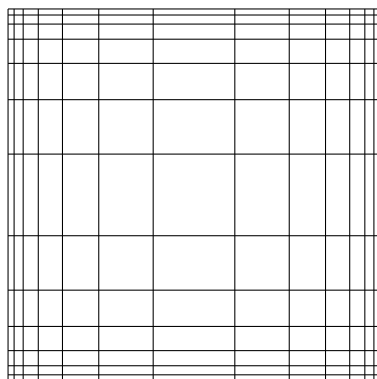
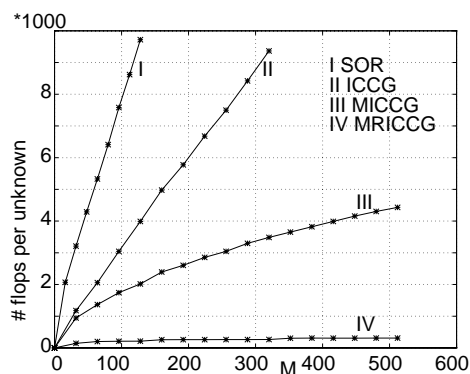
FIG. 3. Diffusion coefficient $a(x, y)$.

FIG. 4. Convergence behavior on an aquifer problem.

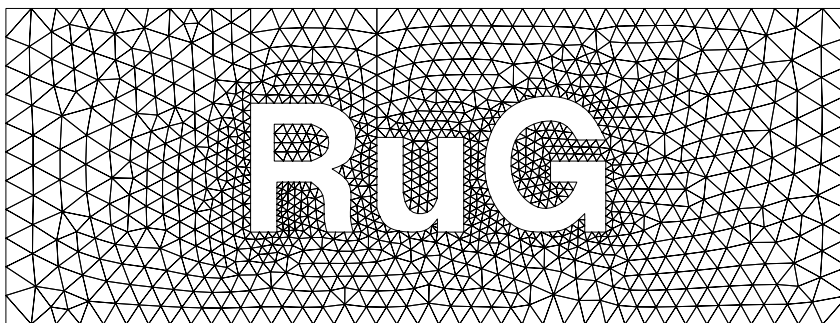
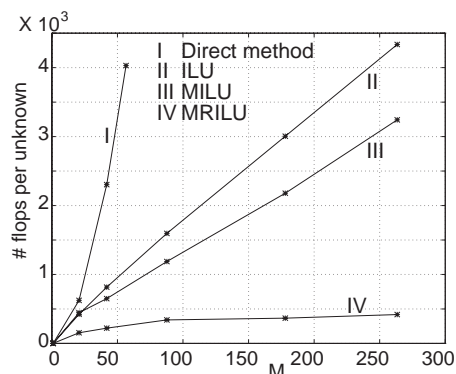
$10^{-6}(u_{max}^{(0)} - u_{min}^{(0)})$. This type of problem may cause the convergence of standard iterative methods (including multigrid) to deteriorate [9]. In Figure 6 matrix renumbering ICCG (MRICCG), the symmetric version of MRILU (with the simplified dropping strategy), is compared to the standard methods SOR, ICCG, and MICCG. The last one again uses a proper perturbation of the diagonal, and for both ICCG and MICCG the efficient Eisenstat implementation is used [25]. Moreover, in MICCG a small perturbation is applied to the diagonal. The figure shows for an $M \times M$ grid the number of flops per unknown as a function of M . Clearly MRILU outperforms the other methods. Moreover, it shows a convergence nearly independent of the grid. All methods show their typical behavior. For SOR it is known that for the optimal overrelaxation factor the convergence is $1 - O(1/M)$ (see [62]) and for ICCG a similar behavior holds. This means that the work is linear in M . For MICCG the amount of work is about \sqrt{M} . This is nicely reflected in the results.

The stretching of the grid may also be interpreted as anisotropy. We applied MRILU to many other types of anisotropic problems, among which are the two-dimensional ones described in [29] and [34]. The behavior is always the same as for

FIG. 5. *Exponentially stretched grid.*FIG. 6. *Comparison with standard iterative methods on a stretched grid.*

the problem described in this section and about 300 flops per grid point are needed. This means that an order of magnitude in efficiency is easily gained with respect to the results in those papers. Similar observations are made in [28].

3.4. Poisson equation on a finite element mesh. MRILU can handle arbitrary sparsity patterns. Here we will show results on a finite element mesh as given in Figure 7. This grid is generated by a finite element package. On this grid $-\Delta u = f$ has been solved with Dirichlet boundary conditions. As stopping criterion we demand that the 2-norm of the residual of the preconditioned system is decreased by at least a factor of 10^6 . Again we compare MRILU, with the simplified dropping strategy, to some standard methods: a direct solver (a frontal method), ILU, and MILU. The results are displayed in Figure 8, where M denotes the square root of the total number of unknowns. As one may expect the direct solver is applicable only to small problems. However, one should note that in the case of a repeated solve direct methods may be attractive. For ILU we see again its predicted linear behavior. We see the same behavior for MILU because the perturbation is difficult to choose on a general mesh. MRILU shows again its nearly grid-independent convergence. Hence, it performs equally well on structured and unstructured grids. For a comparison with other advanced methods on this problem see [6].

FIG. 7. *Finite element grid.*FIG. 8. *Comparison on a finite element grid.*

3.5. Centrally discretized convection-diffusion problem. In this section we will show how MRILU can be applied to convection-dominated problems. We consider the model problem

$$(2) \quad -u_{xx} - u_{yy} + au_x + bu_y = f, \quad a, b \gg 1.$$

In many fluid flow problems convection plays a central role. However, numerically there are some difficulties associated with the terms modeling this phenomenon. For a sufficient accuracy one usually needs at least a second-order discretization. A simple one is the central discretization

$$(3) \quad u_x \approx (u_{j+1} - u_{j-1}) / (2\Delta x).$$

In strongly convection dominated flows this results in weak coupling of odd and even points, which may result in the occurrence of so-called $2\Delta x$ wiggles. For that reason artificial diffusion often is used to restore the coupling.

On sufficient fine grids, nevertheless, central differences are the most accurate [58, 59], but we have to face some problems concerning the coefficient matrix. In strongly convection dominated flows this matrix may be far from an M-matrix. The diagonal is weak with respect to the off-diagonal elements. This is especially difficult

for incomplete decompositions. In complete decompositions partial pivoting usually is used to overcome this problem. However, this destroys the structure in the matrix.

It is also possible to precondition on the basis of the so-called one-sided discretization of the convection-diffusion problem. (For more details on these defect correction type approaches see [20].)

We prefer the use of a block form, as will be explained below. In one dimension there is much similarity between our approach and nested dissection or cyclic reduction. Therefore we will introduce our block form by that method.

One-dimensional cyclic reduction. Consider the stencil

$$\begin{bmatrix} 1 & \varepsilon & -1 \end{bmatrix}$$

as a simplification of the convection-diffusion stencil in one dimension. Eliminating the unknowns at the odd points of the grid results in a system for unknowns at the even points for which the stencil is given by

$$\begin{bmatrix} -1/\varepsilon & \varepsilon + 2/\varepsilon & -1/\varepsilon \end{bmatrix}.$$

To obtain this stencil, multipliers $\pm 1/\varepsilon$ have to be used. In practical computations ε may become as small as 0.01, resulting in large multipliers. However, after this unusual step a symmetric positive definite system occurs which can be solved by standard approaches. Elman and Golub [26] show how such an approach can be utilized.

One-dimensional block cyclic reduction. Suppose we keep unknowns of two subsequent grid points together. We then get a block stencil of the form

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon & -1 \\ 1 & \varepsilon \end{bmatrix} \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}.$$

After elimination of the odd pairs we obtain the following reduced system for the even pairs:

$$\begin{bmatrix} 0 & \frac{1}{1+\varepsilon^2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon + \frac{\varepsilon}{1+\varepsilon^2} & -1 \\ 1 & \varepsilon + \frac{\varepsilon}{1+\varepsilon^2} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ \frac{-1}{1+\varepsilon^2} & 0 \end{bmatrix}.$$

In this case the multipliers used are

$$\begin{bmatrix} \frac{-1}{1+\varepsilon^2} & \frac{\varepsilon}{1+\varepsilon^2} \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 \\ \frac{-\varepsilon}{1+\varepsilon^2} & \frac{-1}{1+\varepsilon^2} \end{bmatrix}.$$

Observe that all elements in the multiplier are now less than one. Hence, the reduction is stable. Furthermore, observe that the diagonal elements of the central block have become larger. As the process is repeated these elements keep growing. (In fact they double as long as they are small with respect to the other elements in the block.) This is a favorable situation which says that the coupling of the unknowns within a pair will become stronger during the reduction and eventually will fix the coupling between odd and even points in the back substitution. Observe that this block approach is applicable for any ε and thus also in cases where it is not necessary.

The extension of this block approach to two dimensions is done as follows. Pairs are chosen on the basis of the dominant flow direction. These pairs are maintained during the whole process. (Note that one could form new pairs for each new Schur complement. This gives slightly better results but is of course also more expensive.)

TABLE 2
Convection-diffusion problem with $a = 10,000$, $b = 1000$.

Problem	Fill	Rel.fill	It.	Flops
S_{CR} $\kappa = 2417$	Low	2.0	35	2390
	Mod.	2.6	11	911
	High	2.8	5	451
S_{BICR} $\kappa = 18$	Low	0.9	8	435
	Mod.	1.4	4	268
	High	1.8	3	230

Convection-diffusion with constant coefficients. We will present some results for the convection-diffusion problem (2) with $a = 10,000$, $b = 1000$. The domain is the unit square; we take Dirichlet conditions at $x = 0$, $y = 0$, and Neumann conditions at $x = 1$, $y = 1$, and improve the preconditioned residual by 6 digits. The number of equidistant grid points in each direction is 32. We choose a convection-dominated flow (mesh Peclet number about 150) and compare the standard approach without blocks, resulting in exactly the same Schur complement as the one used in [26] with the block approach. The results are displayed in Table 2. The first column gives the condition number of the first Schur complement in both approaches and one observes a dramatic difference. The second column gives a fuzzy indication of the fill. This is made more explicit in the third column, where the fill needed for the decomposition relative to that of the original matrix is given. The fourth column shows the number of iterations and the last column gives the flop count for the solution process. Note the relation between the bad conditioning and the iteration. In the pointwise approach far more fill is needed to get an acceptable number of iterations.

Comparison on a rotating flow. For a convection-diffusion problem posed by Morton [37] we will compare our result with those of Elman and Golub [26, Tables 6.1 and 6.2]. In this problem the coefficients of the first derivatives vary and can be interpreted as velocities belonging to a rotating flow. For this problem Elman and Golub use two methods: block Gauss–Seidel and GMRES combined with block ILU. For the choice of the blocks four variants are considered. They are displayed in Figure 9. Our comparison will be expressed in flop counts. In Table 3 the results are shown when the residual is improved by six digits starting with a random vector. Here (2,M–) denotes that a block size of 2 has been used with the unmodified approach (M+ for modified). For $1/\varepsilon = 100$ the matrix is almost triangular, which explains the good convergence for all methods. For $1/\varepsilon = 1000$ the block variant performs much better than the other methods.

3.6. SHERMAN problems. Recently, Chapman, Saad, and Wigton [17] showed results for incomplete decompositions on the SHERMAN problems 2, 3, and 5.² In this section we compare these results with those of MRILU. (For problems 1 and 4 MRILU performs analogously.) In Table 4 these problems are described briefly. In the comparison we adopted results from [17] of GMRES(50) with various preconditioners:

- ILUT: threshold drop tolerance and fill number;
- ILUD: Gustafsson’s modification variant of ILUT; drop tolerance only;
- ILU(k): dropping strategy based on “level of fill.”

²SHERMAN problems are available as part of the Harwell–Boeing Collection at <http://math.nist.gov:80/MatrixMarket>. Users should be aware that the block size in SHERMAN2 is 6 instead of the given number 5.

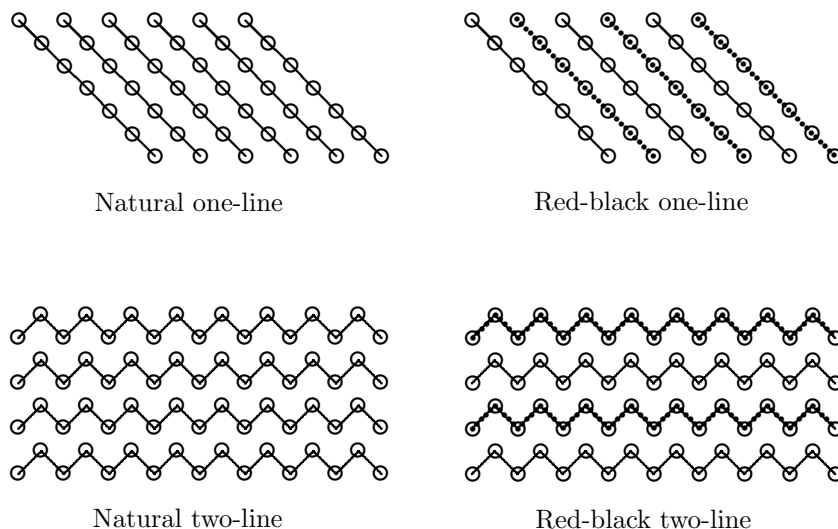


FIG. 9. Blocks used in ILU and Gauss-Seidel.

It is interesting to compare our approach with these methods. The methods are quite similar but MRILU adds the matrix-dependent ordering and the use of blocks. A random right-hand side is used with a zero starting vector. The residual is improved by eight digits. The results can be found in Table 5. For the SHERMAN2 problem one observes that ILUD has serious convergence problems. It is well known (see section 1) that a modified ILU factorization is more sensitive to convergence problems than its unmodified ILU variant. Here the unmodified ILUT performs reasonably only with high fill. This is because one solves a problem with block size 6 by a pointwise method. A pointwise version of MRILU has similar difficulties with this problem.

The SHERMAN3 problem looks like a Poisson problem. The matrix is almost symmetrical and it turns out that here MRILU can also be combined with CG. Compared to Bi-CGSTAB, this doubles the number of iterations but also halves the number of matrix vector multiplications. Therefore the number of flops is comparable.

The ordering produced by MRILU and the use of blocks for SHERMAN2 clearly has a favorable influence on the results. An order of magnitude is easily gained for a comparable fill.

3.7. The incompressible Navier–Stokes equations. In this section some preliminary results for the two-dimensional incompressible Navier–Stokes equations are presented. There is some resemblance to the AMG approach followed by Webster [60].

The incompressible Navier–Stokes equations read

$$(4) \quad \begin{aligned} u_t &= -uu_x - vu_y - p_x + \frac{1}{\text{Re}}(u_{xx} + u_{yy}), \\ v_t &= -uv_x - vv_y - p_y + \frac{1}{\text{Re}}(v_{xx} + v_{yy}), \end{aligned}$$

$$u_x + v_y = 0$$

TABLE 3
Comparison on a rotating flow.

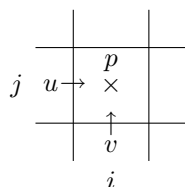
31 × 31 grid							
Method		1/ε					
		10		100		1000	
	Ordering	r.fill	flops	r.fill	flops	r.fill	flops
Gauss–Seidel	Nat. one-line		1098		243		>1350
	RB one-line		1071		261		>1350
	Nat. two-line		1026		234		>1350
	RB two-line		999		234		>1350
GMRES/ILU	Nat. one-line		260		182		858
	RB one-line		1071		702		1924
	Nat. two-line		364		260		2262
	RB two-line		624		676		1846
Fill		(1,M+)		(1,M+)		(2,M−)	
Bi-CGSTAB/MRILU	Low	1.0	303	1.1	201	1.5	482
	Mod.	1.4	238	1.5	165	2.4	434
	High	1.9	198	1.8	136	3.8	321

63 × 31 grid							
Gauss–Seidel	Nat. 1-line		>1350		198		>1350
	Nat. 2-line		1161		198		>1350
GMRES/ILU	Nat. 1-line		442		260		>3900
	Nat. 2-line		520		312		1664
Fill		(1,M+)		(1,M+)		(2,M−)	
Bi-CGSTAB/MRILU	Low	1.1	333	1.1	210	1.5	417
	Mod.	1.5	271	1.5	170	2.3	335
	High	2.0	237	1.8	154	3.4	307

TABLE 4
Short description of SHERMAN problems.

Problem	Order	Nonzeros	Description
SHERMAN2	1080	23094	Thermal simulation, steam injection
SHERMAN3	5005	20033	Black oil, IMPES simulation
SHERMAN5	3312	20793	Fully implicit black oil simulator

with u and v the horizontal and vertical velocity, p the pressure, and Re the Reynolds number. They are discretized using a finite-difference formulation on a uniform staggered grid:



The diffusion term is discretized using the standard second-order central scheme. For the convection term a first-order upwind scheme is used. The equations are solved simultaneously; therefore we can keep together the unknowns belonging to a single grid cell in a vector $w_{i,j} = (u_{i-\frac{1}{2},j}, v_{i,j-\frac{1}{2}}, p_{i,j})$ and denote the discretized system with the stencil

$$\begin{array}{ccc} NW & N & \\ W & C & E \\ & S & SE, \end{array}$$

TABLE 5
Flop count per unknown for SHERMAN problems 2, 3, and 5.

Method	Fill	SHERMAN2			SHERMAN3			SHERMAN5		
		r.fill	it.	flops	r.fill	it.	flops	r.fill	it.	flops
ILUD	Low	0.8	n.c.	n.c.	1.0	77	8150	1.4	36	3905
	High	1.6	76	15127	1.6	32	2907	4.0	17	1744
ILUT	Low	0.7	145	25504	0.9	216	25174	1.0	30	2754
	High	1.5	12	1661	2.1	46	5634	1.9	21	1772
ILU(k)	k=0	1.0	45	8171	1.0	233	27316	1.0	36	3714
	k=2	3.0	7	1327	3.1	50	6746	3.1	19	1810
		(6,M-)			(1,M+)			(1,M+)		
MRILU	Low	0.4	4	634	1.0	19	747	0.6	8	380
	Mod.	0.5	3	490	1.3	12	518	0.8	6	315
	High	0.7	2	391	1.9	9	450	1.1	4	230

TABLE 6
Results for the incompressible Navier-Stokes equations.

Problem	Fill	Rel.fill	It.	Flops
Steady state	Low	2.7	14	1723
	Mod.	3.4	11	1557
	High	4.1	9	1471
Time dependent	Low	1.1	13	1040
	Mod.	1.7	11	1067
	High	2.1	9	982

where the coefficients are 3×3 matrices.

In Table 6 we show results of the solution of a typical linear system coming about in the Newton iteration in the steady state driven cavity problem; see [54] for more detail. The problem is discretized on a 33×33 grid (3267 unknowns) and the preconditioned residual is improved by six digits. For the time-dependent problem the time step is such that a Courant number of approximately 1 is obtained. One observes that the steady state problem asks for a much higher fill than the time-dependent problem.

4. Conclusions. Preconditioning techniques combined with CG-like iterations methods provide powerful tools for solving large sparse systems of equations. Motivated by the success of the nested grid method NGILU, we developed the more general MRILU method, which also can be used on unstructured grids.

An attractive property of this method is that its structure is simple: it is merely an ILU factorization. The essential ingredients are the ordering, which is carried out during the factorization process, and the dropping, which has to be done carefully in order not to destroy the convergence. Both ordering and dropping are based on the size of the entries of the matrix.

Convergence behavior is observed that is nearly independent of the mesh size, an attractive property for very large problems. The method has been applied successfully to symmetric, nonsymmetric, and indefinite problems. For many of the problems shown in this paper the method decreases the number of flops needed to get a prescribed accuracy by an order of magnitude in comparison with other advanced iteration methods. Implementations on high-performance computers are possible due to the high degree of independence in the L and U factors.

The analysis of such a general method is difficult and will be the subject of future research. It may be expected that this will lead to further improvements of the method.

Acknowledgment. The authors wish to thank Prof. Henk van der Vorst for providing the code for the discretization of the aquifer problem.

REFERENCES

- [1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, London, 1994.
- [2] O. AXELSSON AND V. EIJKHOUT, *The nested recursive two-level factorization method for nine-point difference matrices*, SIAM J. Sci. Statist. Comput., 12 (1991), pp. 1373–1400.
- [3] O. AXELSSON AND N. MUNKSGAARD, *Analysis of incomplete factorizations with fixed storage locations*, in Preconditioning Methods, Analysis and Applications, D.J. Evans, ed., Gordon and Breach, London, 1983, pp. 219–241.
- [4] R.E. BANK AND C. WAGNER, *Multilevel ILU Decomposition*, Numer. Math., to appear.
- [5] R. BEAUWENS AND L. QUENON, *Existence criteria for partial matrix factorizations in iterative methods*, SIAM J. Numer. Anal., 13 (1976), pp. 615–643.
- [6] E.F.F. BOTTA, K. DEKKER, Y. NOTAY, A. VAN DER PLOEG, C. VUIK, F.W. WUBS, AND P.M. DE ZEEUW, *How fast the Laplace equation was solved in 1995*, Appl. Numer. Math., 24 (1997), pp. 439–455.
- [7] E.F.F. BOTTA AND A. VAN DER PLOEG, *Preconditioning techniques for matrices with arbitrary sparsity patterns*, in Proceedings of the Ninth International Conference on Finite Elements in Fluids, New Trends and Applications, University of Padova, Venice, 1995, pp. 989–998.
- [8] E.F.F. BOTTA, A. VAN DER PLOEG, AND F.W. WUBS, *Nested grids ILU-decomposition (NGILU)*, J. Comput. Appl. Math., 66 (1996), pp. 515–526.
- [9] E.F.F. BOTTA AND F.W. WUBS, *The convergence behaviour of iterative methods on severely stretched grids*, Internat. J. Numer. Methods Engrg., 36 (1993), pp. 3333–3350.
- [10] E.F.F. BOTTA, F.W. WUBS, AND A. VAN DER PLOEG, *A fast linear-system solver for large unstructured problems on a shared-memory computer*, in Proceedings of the Conference on Algebraic Multilevel Iteration Methods with Applications, O. Axelsson and B. Polman, eds., University of Nijmegen, Nijmegen, The Netherlands, 1996, pp. 105–116.
- [11] J. BRAMBLE, J. PASCIAK, AND J.-C. XU, *Parallel multilevel preconditioners*, Math. Comp., 55 (1990), pp. 1–22.
- [12] CL.W. BRAND, *An incomplete-factorization preconditioning using red-black ordering*, Numer. Math., 61 (1992), pp. 433–454.
- [13] N.I. BULEEV, *A numerical method for the solution of two-dimensional and three-dimensional equations of diffusion*, Mat. Sb., 51 (1960), pp. 227–238.
- [14] J.R. BUNCH AND B.N. PARLETT, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.
- [15] T.F. CHAN AND H.A. VAN DER VORST, *Approximate and incomplete factorizations*, Parallel Numerical Algorithms, ICASE/LaRC Interdiscip. Ser. Sci. Eng. 4, Kluwer Academic Publishers, Dordrecht, the Netherlands, 1997, pp. 167–202.
- [16] Q. CHANG, Y.S. WONG, AND H. FU, *On the algebraic multigrid method*, J. Comput. Phys., 125 (1996), pp. 279–292.
- [17] A. CHAPMAN, Y. SAAD, AND L. WIGTON, *High-order ILU Preconditioners for CFD Problems*, Technical report UMSI 96/14, University of Minnesota, Minneapolis, 1996.
- [18] E.F. D’AZEVEDO, P.A. FORSYTH, AND W.P. TANG, *Towards a cost-effective ILU preconditioner with high level fill*, BIT, 32 (1992), pp. 442–463.
- [19] P.M. DE ZEEUW, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 3 (1990), pp. 1–27.
- [20] J.-A. DÉSIDÉRI AND P.W. HEMKER, *Convergence analysis of the defect-correction iteration for hyperbolic problems*, SIAM J. Sci. Comput., 16 (1995), pp. 88–118.
- [21] I.S. DUFF, A.M. ERISMAN, AND J.K. REID, *Direct Methods for Sparse Matrices*, Monogr. Numer. Anal., Oxford University Press, New York, 1986.
- [22] I.S. DUFF AND G.A. MEURANT, *The effect of ordering on preconditioned conjugate gradients*, BIT, 29 (1989), pp. 635–657.
- [23] T. DUPONT, R.P. KENDALL, AND H.H. RACHFORD, JR., *An approximate factorization procedure for solving self-adjoint elliptic difference equations*, SIAM J. Numer. Anal., 5 (1968), pp. 559–573.
- [24] V. EIJKHOUT, *Beware of unperturbed modified incomplete factorizations*, in Iterative Methods in Linear Algebra, R. Beauwens and P. de Groen, eds., North-Holland, Amsterdam, 1992, pp. 583–591.
- [25] S.C. EISENSTAT, *Efficient implementation of a class of preconditioned conjugate gradient methods*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 1–4.

- [26] H.C. ELMAN AND G.H. GOLUB, *Line iterative methods for cyclically reduced discrete convection-diffusion problems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 339–363.
- [27] G.H. GOLUB AND C.F. VAN LOAN, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, MD, 1996.
- [28] T. GRAUSCHOPF, M. GRIEBEL, AND H. REGLER, *Additive multilevel preconditioners based on bilinear interpolation, matrix dependent geometric coarsening and algebraic multigrid coarsening for second order elliptic PDEs*, Appl. Numer. Math., 23 (1997), pp. 63–95.
- [29] M. GRIEBEL AND P. OSWALD, *Tensor-product-type subspace splittings and multilevel iterative methods for anisotropic problems*, Adv. Comput. Math., 4 (1995), pp. 171–206.
- [30] I. GUSTAFSSON, *A class of 1st order factorization methods*, BIT, 18 (1978), pp. 142–156.
- [31] W. HACKBUSCH, *Iterative Solution of Large Sparse Linear Systems of Equations*, Appl. Math. Sci. 95, Springer-Verlag, New York, 1994.
- [32] E.F. KAASSCHIETER, *Preconditioned conjugate gradients for solving singular systems, Iterative methods for the solution of linear systems*, J. Comput. Appl. Math., 24, (1988), pp. 265–275.
- [33] S. KNAPEK, *Matrix-dependent multigrid homogenization for diffusion problems*, SIAM J. Sci. Comput., 20 (1999), pp. 515–533.
- [34] C.-C.J. KUO, T.F. CHAN, AND C. TONG, *Multilevel filtering elliptic preconditioners*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 403–429.
- [35] J.A. MEIJERINK AND H.A. VAN DER VORST, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.
- [36] M.-M. MAGOLU, *Ordering strategies for modified block incomplete factorizations*, SIAM J. Sci. Comput., 16 (1995), pp. 378–399.
- [37] K.W. MORTON, *Generalised Galerkin methods for steady and unsteady problems*, in Numerical Methods for Fluid Dynamics, K.W. Morton and M.J. Baines, eds., Academic Press, New York, 1982, pp. 1–32.
- [38] Y. NOTAY, *Solving positive (semi)definite linear systems by preconditioned iterative methods*, in Preconditioned Conjugate Gradient Methods, O. Axelsson and L.Y. Kolotilina, eds., Lecture Notes in Math. 1457, Springer-Verlag, Berlin, 1990, pp. 105–125.
- [39] Y. NOTAY, *Conditioning analysis of modified block incomplete factorizations*, Linear Algebra Appl., 154/156 (1991), pp. 711–722.
- [40] Y. NOTAY, *DRIC: A dynamic version of the RIC method*, J. Numer. Linear Algebra Appl., 1 (1994), pp. 511–532.
- [41] Y. NOTAY AND Z. OULD AMAR, *A nearly optimal preconditioning based on recursive red-black orderings*, J. Numer. Linear Algebra Appl., 4 (1997), pp. 369–391.
- [42] T.A. OLIPHANT, *An implicit numerical method for solving two-dimensional time-dependent diffusion problems*, Quart. Appl. Math., 19 (1961), pp. 221–229.
- [43] T.A. OLIPHANT, *An extrapolation process for solving linear systems*, Quart. Appl. Math., 20 (1962), pp. 257–267.
- [44] A. REUSKEN, *Multigrid with matrix-dependent transfer operators for a singular perturbation problem*, Computing, 50 (1993), pp. 199–212.
- [45] A. REUSKEN, *Fourier analysis of a robust multigrid method for convection-diffusion equations*, Numer. Math., 71 (1995), pp. 365–398.
- [46] A. REUSKEN, *On the approximate cyclic reduction preconditioner*, SIAM J. Sci. Comput., to appear.
- [47] J.W. RUGE AND K. STÜBEN, *Algebraic multigrid*, in Multigrid Methods, S.F. McCormick, ed., Frontiers Appl. Math. 3, SIAM, Philadelphia, PA, 1987, pp. 73–130.
- [48] Y. SAAD, *Highly Parallel Preconditioners for General Sparse Matrices*, Recent Advances in Iterative Methods, IMA Vol. Math. Appl. 60, Springer-Verlag, New York, 1994, pp. 165–199.
- [49] Y. SAAD, *ILUM: A multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput., 17 (1996), pp. 830–847.
- [50] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, MA, 1996.
- [51] Y. SAAD AND J. ZHANG, *BILUM: Block Versions of Multi-Elimination and Multi-Level ILU Preconditioner for General Sparse Linear Systems*, Technical report UMSI 97-126, University of Minnesota, Minneapolis, 1997.
- [52] G.L.G. SLEIJPEN AND H.A. VAN DER VORST, *Maintaining convergence properties of BiCGstab methods in finite precision arithmetic*, Numer. Algorithms, 10 (1995), pp. 203–223.
- [53] G.L.G. SLEIJPEN AND H.A. VAN DER VORST, *Reliable updated residuals in hybrid Bi-CG methods*, Computing, 56 (1996), pp. 141–163.
- [54] G. TIESINGA, *Block preconditioned BiCGstab(2) for solving the Navier-Stokes equation*, Z.

- Angew. Math. Mech., 76 (1996), pp. 563–564.
- [55] A. VAN DER PLOEG, *Preconditioning techniques for large sparse, non-symmetric matrices with arbitrary sparsity patterns*, in Iterative Methods in Linear Algebra, R. Beauwens and P. de Groen, eds., North-Holland, Amsterdam, 1992, pp. 173–179.
 - [56] A. VAN DER PLOEG, *Preconditioning for Sparse Matrices with Applications*, Ph.D. thesis, University of Groningen, Groningen, The Netherlands, 1994.
 - [57] H.A. VAN DER VORST, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644.
 - [58] R.W.C.P. VERSTAPPEN AND A.E.P. VELDMAN, *A fourth-order finite volume method for direct numerical simulation of turbulence at higher Reynolds numbers*, in Computational Fluid Dynamics '96, J. Periaux, et al., eds., John Wiley, New York, 1996, pp. 1073–1079.
 - [59] R.W.C.P. VERSTAPPEN AND A.E.P. VELDMAN, *Direct numerical simulation of turbulence at lower costs*, J. Engrg. Math., 32 (1997), pp. 143–159.
 - [60] R. WEBSTER, *An algebraic multigrid solver for Navier-Stokes problems in the discrete second-order approximation*, Internat. J. Numer. Methods Fluids, 22 (1996), pp. 1103–1123.
 - [61] J. XU, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.
 - [62] D.M. YOUNG, *Iterative Solution of Large Linear Systems*, Academic Press, New York, 1971.